UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/606,786	06/28/2000	Robert Murphy	34918.0100	6327

20322 7590 05/12/2004

SNELL & WILMER
ONE ARIZONA CENTER
400 EAST VAN BUREN
PHOENIX, AZ 850040001

EXAMINER	
CHOUDHURY, AZIZUL Q	
ART UNIT	**PAPER NUMBER**
2143	6

DATE MAILED: 05/12/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/606,786	MURPHY ET AL.	
	Examiner	Art Unit	
	Azizul Choudhury	2143	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *05 March 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-116* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-116* is/are rejected.

7)☒ Claim(s) *96-97* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *28 June 2000* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date 3.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Specification

1. The disclosure is objected to because of the following informalities:

   i. The use of trademarked terminology requires the use of the TM emblem. The following are just some of the TM omissions discovered by the office.

      (1) The term "Microsoft" in line 2, page 2 should be "Microsoft™".

      (2) The term "Symantec" in lines 17, page 2 should be "Symantec™".

      (3) The term "Mac OS" in line 8, page 6 should be "Mac OS™".

      (4) The term "Appletalk" in line 13, page 6 should be "Appletalk™".

   ii. This list does not indicate every TM omission; it simply serves to better illustrate the issue. Appropriate correction of all the TM omitted terminology is required.

Claim Objections

1. The following claims are objected to for the reasons listed below.

2. Claims 96 and 97 are objected to because of the following informalities:

iii. The use of trademarked terminology requires the use of the TM emblem. The following are just some of the TM omissions discovered by the office.

(5) The term "Netware" in claim 96 should be "Netware™".

(6) The term "Microsoft" in claim 97 should be "Microsoft™".

This list does not indicate every TM omission; it simply serves to better illustrate the issue. Appropriate correction of all the TM omitted terminology is required.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 1-20, 42-49, 51, 62-69, 75-82, 86, 89, 90, 98-101 and 105-115 are rejected under 35 U.S.C. 102(b) as being anticipated by Chang (US5680547A), hereafter referenced as Chang.

4. With regards to claim 1, Chang teaches a method comprising the steps of:

- Receiving said attributes from said client computer at a server computer prior to said client computer loading said local operating system (Chang discloses a design that allows client machines to communicate with a server, prior to

loading the operating system. This communication ensures that server management applications can run properly (column 2, lines 44-54, Chang). For a server management application to function properly, it must have the claimed management instructions for the client machine);

- Selecting one of a plurality of management instructions for said client computer at said server computer, wherein said one of a said plurality of management instructions is selected based upon said attributes of said client computer (As stated above, Chang's design consists of a server management application. For such an application to function properly, it is inherent that the application contains the claimed management instructions, as well as selects the appropriate management instructions for each client machine. If this step did not exist, a server management application could never properly perform its task); and
- Providing said one of said plurality of management instructions from said server computer to said client computer (As stated above, Chang's design provides for communication between the server and the client machine. This communication is the means by which the server management application can manage the client machine. Hence, the communication is used to send management instructions from the server to the client as claimed) to thereby allow said client computer to execute said one of said plurality of management instructions at said client computer (Chang's design consists of server management applications. For server management applications to

function properly, management instructions sent by the server management applications must execute on the client machines).

5. With regards to claim 2, Chang teaches that the step of selecting said one of said plurality of management instructions comprises:

- Providing a configuration determination application from the server to the client computer prior to the receiving step (First, means by which to perform such a task is present in Chang's design. Remote software installation and distribution to client machines is present in Chang's design (column 3, lines 31-33, Chang). In addition, Chang further states that files can be transferred from the server to the workstation (column 2, lines 66-67, Chang). This just further shows that programs can be sent from the server to the workstation. It is inherent that the server must provide an application to the client prior to the receiving because there is no other way for a client to receive data over a network. Anyone within the networking and computing field is informed of the fact that for data transfers to occur in a network (including the receipt of commands), both the server and the client, each must have an initial software and/or hardware to allow for such tasks. In Chang's design, this initial software/hardware (application) present within the client prior to the receiving step is the firmware);
- Determining said attributes of said client computer with the configuration determination application (Chang's design permits the server with the server

management application to meter and diagnose client machines (column 3, lines 31-33, Chang). Having the ability to diagnose and meter clients is equivalent to having the ability to determine a client's attributes); and

- Providing said attributes from said configuration determination application to said server computer (The server management application of Chang's design is contained within a server. Hence, the information obtained by the server management application is available to the server).

6. With regards to claim 3, Chang teaches that the obtained attributes comprise hardware attributes (Chang discloses a design that features client machines with hardware (column 4, lines 10-12, Chang). These can be accessed by the server management application of Chang's design. Since the server management application can access the client machine, some attributes of the client must be known. These attributes can comprise of hardware attributes as claimed).

7. With regards to claim 4, Chang teaches that the obtained attributes comprise firmware attributes (Chang's design has client machines with firmware (column 4, lines 13-15, Chang). These can be accessed by the server management application of Chang's design hence, the attributes can comprise of firmware attributes as claimed).

8. With regards to claim 5, Chang teaches that the obtained attributes comprise DMI attributes (Chang's design has client machines with NICs to allow for interfacing

with server management applications (column 4, lines 13-15, Chang). Furthermore, Chang's design consists of a hardware device attached to each client machine to permit access to the client from the desktop management software at any given time (column 2, lines 38-54, Chang). These can be accessed by the server management application of Chang's design hence, the attributes can comprise of DMI attributes as claimed).

9. With regards to claim 6, Chang teaches that the obtained attributes comprise PCI attributes (PCI is a type of hardware and Chang's design has client machines with hardware (column 4, lines 10-12, Chang). These can be accessed by the server management application of Chang's design hence, the attributes can comprise of PCI attributes as claimed).

10. With regards to claim 7, Chang teaches that the obtained attributes comprise SMBIOS attributes (SMBIOS is a type of BIOS. Chang's design allows for the server management application within a server to access a client's BIOS (column 2, line 48, Chang). For the server management application to access the BIOS properly, it must have some means by which to obtain the attributes of the BIOS. Since SMBIOS and BIOS are viewed as being equivalent, the server management application must be able to obtain the SMBIOS attributes).

11. With regards to claim 8, Chang teaches that the obtained attributes comprise at least one of the group consisting of system manufacturer, model, motherboard type, bus

information, and adapter information (Chang's design allows for the server management application to access the motherboard (column 2, lines 38-44, Chang). Since the motherboard can be accessed, its attributes can be obtained hence, the motherboard type can be obtained. In addition for server management applications to function properly, attributes of the client must be obtained; any of the attributes claimed are reasonable attributes that could be claimed by many server management applications).

12. With regards to claim 9, Chang teaches a method where said adapter information comprises information about adapter orientation within a system bus of said client computer (For a server management application within the server in Chang's design to function properly, it must know details concerning the client machines it is dealing with. Chang's disclosure states that an advantage of his design is the ability for "remote software installation, distribution, metering and diagnostics." In addition, it offers the ability for configuration management. Chang's disclosure further states the design offers "workstation recovery" and "critical node monitoring" (column 3, lines 25-33, Chang). With a design that allows for the server management application within the server to monitor such various aspects of the client, it is inherent that the client's adapter information (including that comprising information about the adapter's orientation within a system bus) would be available to the server management application as well).

13. With regards to claim 10, Chang teaches a method where the client computer comprises a file system and wherein the method further comprises the step of verifying said file system of said client computer (Operating systems contain file systems and Chang's design calls for client machines to have operating systems (column 2, line 20, Chang). In addition, Chang's design allows for workstation crash recovery (column 3, line 31, Chang). For workstation crash recovery to function properly, files within the client must be checked against some form of index of files. Hence the claimed verification process occurs).

14. With regards to claim 11, Chang teaches a method where the verifying of the file system comprises checking the files in said file system against an index file (Operating systems contain file systems and Chang's design calls for client machines to have operating systems (column 2, line 20, Chang). In addition, Chang's design allows for workstation crash recovery (column 3, line 31, Chang). For workstation crash recovery to function properly, files within the client must be checked against some form of index of files. Hence the claimed index file must exist).

15. With regards to claim 12, Chang teaches a method where the index file is retained on the server computer and the step of verifying the file system is executed on the server computer (Operating systems contain file systems and Chang's design calls for client machines to have operating systems (column 2, line 20, Chang). In addition, Chang's design allows for workstation crash recovery (column 3, line 31, Chang). For

workstation crash recovery to function properly, it is inherent that files within the client must be checked against some form of index of files. For the recovery of a client's files to occur, the index files must be stored within some storage, which is accessible for copying when needed (such as within the server or client). Furthermore the server management application within the server handles the process of installation, distribution, metering and diagnostics (column 3, lines 31-32, Chang) (as stated above, software is comprised of files and hence processes performed on software can be performed on files). Metering and diagnostics are viewed as being equivalent to verification and hence, the verification of the file system are performed on the server of Chang's design as claimed. Hence the claimed method exists).

16. With regards to claim 13, Chang teaches a method where the index file is retained on the client computer and the step of verifying the file system is executed on the client computer (Operating systems contain file systems and Chang's design calls for client machines to have operating systems (column 2, line 20, Chang). In addition, Chang's design allows for workstation crash recovery (column 3, line 31, Chang). For workstation crash recovery to function properly, it is inherent that files within the client be checked against some form of index of files. For the recovery of a client's files to occur, the index files must be stored within some storage, which is accessible for copying when needed (such as within the server or client). Furthermore, it is well known in the art that operating systems have the capability to verify files and the file system).

17. With regards to claim 14, Chang teaches a method where the index file is compressed (File compression requires a file to be rewritten in a different format. Chang's design discloses that an ID (a file) can be encoded (column 2, line 55, Chang). Encoding like compression requires a file to be rewritten in a different format. It would therefore have been inherent to use file compression).

18. With regards to claim 15, Chang teaches a method where files missing from said file system are retrieved from said server computer (Chang's design allows for workstation crash recovery (column 3, line 31, Chang). Workstation crash recovery is a process by which missing files are restored to the client machine by the server management application within the server. Hence missing files are retrieved from the server to the client as claimed).

19. With regards to claim16, Chang teaches a method where the index file corresponds to the attributes of the client computer (Index files are used as a reference to maintain the correctness of files. In this case the index file serves as a reference to files in client machines. If files serve as index files to a client, it is inherent that the files would possess not only information pertaining to the files within the client but also possess information pertaining to the client machine itself. The claim is therefore rejected).

20. With regards to claim 17, Chang teaches a method where files are accessed using the PXE protocol (Chang's design teaches that the server management application can perform pre-boot tasks on the client machine (column 2, lines 65-66, Chang). To perform any task, a type of environment must exist (even for pre-boot tasks). Since the PXE protocol is a pre-boot execution environment, it is viewed as being equivalent to Chang's design which is capable of performing (executing) pre-boot tasks).

21. With regards to claim 18, Chang teaches a method where the contacting step is in accordance with the PXE protocol (For any contact step to occur, it is inherent for the step to abide by some protocol as is used for the file accessing. In addition, Chang's design teaches that the server management application can perform pre-boot tasks on the client machine (column 2, lines 65-66, Chang). To perform any task, a type of environment must exist (even for pre-boot tasks). Since the PXE protocol is a pre-boot execution environment, it is viewed as being equivalent to Chang's design which is capable of performing (executing) pre-boot tasks).

22. With regards to claim 19, Chang teaches a method comprising the step of mounting a remote drive from the server computer to the client computer (Chang's design calls for both server and client machines. The design enables the client and server to communicate between each other (column 4, lines 23-24, Chang). It is inherent that both the server and client machines have drives such as hard drives.

Since a client can communicate with a server (which is remote), it is possible for a client to access a server's hard drive. This accessing of a server's hard drive requires mounting and hence the claimed ability to mount a remote drive from the server to the client is viewed as being inherent).

23. With regards to claim 20, Chang teaches a method where the step of executing the management instructions comprises accessing data files on the remote drive (As stated before, Chang's design allows for a server management application within a server. It is inherent that a server has hard drives (which is a type of drive) by which to store data files. This setup is equivalent to the claimed data files located on a remote drive. It was also stated before that Chang's design allows for clients and servers to communicate with one another (column 4, lines 23-24, Chang). This arrangement permits the data files within the server (remote drive) to be accessed as claimed).

24. With regards to claim 42, Chang teaches a method comprising the steps of:

- Providing a configuration determination program from a server in response to a request from said client computer, said configuration determination program being configured to identify attributes of said client computer and to provide said attributes to said server (Chang's design offers the ability for the client to communicate with the server with the server management application within it (column 2, lines 48-52, Chang). This allows for the client to make requests to the server and to have the server respond to a request as claimed. The

server management application program within the server can also send updates for the boot files prior to the client machine booting (column 2, lines 65-66, Chang). Such boot file updates is viewed as being equivalent to the tasks performed by the claimed boot configuration program);

- Receiving said attributes from said configuration determination program at said server (Chang's design offers the ability for the client to communicate with the server with the server management application within it (column 2, lines 48-52, Chang). This allows for the client to make requests to the server and to have the server respond to a request as claimed);
- Selecting one of a plurality of management instructions for said client computer at said server as a function of said attributes (As stated above, Chang's design consists of a server management application. For such an application to function properly, it is inherent that it not only contains the claimed management instructions, but also that it select the appropriate management instructions for each client machine. If this step did not exist, a server management application could never properly perform it's task); and
- Providing said one of said plurality of management instructions from said server to said client computer prior to booting said client computer (As stated above, Chang's design provides for communication between the server and the client machine. This communication is the means by which the server management application can manage the client machine. Hence, the

communication is used to send management instructions from the server to the client as claimed).

25. With regards to claim 43, Chang teaches a method where the attributes comprise hardware attributes (Chang's design has client machines with hardware (column 4, lines 10-12, Chang). These can be accessed by the server management application of Chang's design hence, the attributes can comprise of hardware attributes as claimed).

26. With regards to claim 44, Chang teaches a method where the attributes comprise firmware attributes (Chang's design has client machines with firmware (column 4, lines 13-15, Chang). These can be accessed by the server management application of Chang's design hence, the attributes can comprise of firmware attributes as claimed).

27. With regards to claim 45, Chang teaches a method comprising the step of executing said one of a plurality of management instructions at said client computer (Chang's design consists of server management applications. For server management applications to function properly, management instructions sent by the server management applications must execute on the client machines).

28. With regards to claim 46, Chang teaches a method wherein said one of said plurality of management instructions comprises at least one of a plurality of scripts (Chang discloses a design which allows for scripting languages to be used to program

client-server operations (column 4, lines 33-35, Chang). Client-server operations are viewed as being equivalent to management instructions).

29. With regards to claim 47, Chang teaches a method wherein at least one of said plurality of scripts is a REXX script (Chang discloses a design which allows for scripting languages to be used to program client-server operations (column 4, lines 33-35, Chang). REXX is a scripting language and hence the claim is rejected).

30. With regards to claim 48, Chang teaches a method wherein at least one of said plurality of scripts is a PERL script (Chang discloses a design which allows for scripting languages to be used to program client-server operations (column 4, lines 33-35, Chang). PERL is a scripting language and hence the claim is rejected).

31. With regards to claim 49, Chang teaches a method wherein at least one of said plurality of scripts is a batch script (Chang discloses a design which allows for scripting languages to be used to program client-server operations (column 4, lines 33-35, Chang). Chang stated no limitations and a batch script is still a form of a scripting language. Hence, the claim is rejected).

32. With regards to claim 51, Chang teaches a method wherein each script comprises instructions for executing one or more tasks in response to the occurrence of at least one event (Chang discloses a design which allows for scripting languages to be

used to program client-server operations (column 4, lines 33-35, Chang). Scripts are programmed to perform desired tasks. With that in mind, it is inherent that scripts are capable of responding to events as claimed).

33. With regards to claim 62 Chang teaches a method wherein said client computer comprises a file system and wherein said step of managing said client computer comprises verifying said file system of said client computer (Operating systems contain file systems and Chang's design calls for client machines to have operating systems (column 2, line 20, Chang). In addition, Chang's design allows for workstation crash recovery (column 3, line 31, Chang). For workstation crash recovery to function properly, files within the client must be checked against some form of index of files. Hence the claimed verification process occurs).

34. With regards to claim 63, Chang teaches a method wherein said step of verifying said file system comprises checking the files in said file system against an index file (Operating systems contain file systems and Chang's design calls for client machines to have operating systems (column 2, line 20, Chang). In addition, Chang's design allows for workstation crash recovery (column 3, line 31, Chang). For workstation crash recovery to function properly, files within the client must be checked against some form of index of files. Hence the claimed index file must exist).

35. With regards to claim 64, Chang teaches a method wherein said index file is retained on said server computer and wherein said step of verifying said file system is executed on said server computer (Operating systems contain file systems and Chang's design calls for client machines to have operating systems (column 2, line 20, Chang). In addition, Chang's design allows for workstation crash recovery (column 3, line 31, Chang). For workstation crash recovery to function properly, it is inherent that files within the client must be checked against some form of index of files. For the recovery of a client's files to occur, the index files must be stored within some storage, which is accessible for copying when needed (such as within the server or client). Furthermore the server management application within the server handles the process of installation, distribution, metering and diagnostics (column 3, lines 31-32, Chang) (as stated above, software is comprised of files and hence processes performed on software can be performed on files). Metering and diagnostics are viewed as being equivalent to verification and hence, the verification of the file system are performed on the server of Chang's design as claimed. Hence the claimed method exists).

36. With regards to claim 65, Chang teaches a method wherein said index file is retained on said client computer and wherein said step of verifying said file system is executed on said client computer (Operating systems contain file systems and Chang's design calls for client machines to have operating systems (column 2, line 20, Chang). In addition, Chang's design allows for workstation crash recovery (column 3, line 31, Chang). For workstation crash recovery to function properly, it is inherent that files

within the client must be checked against some form of index of files. For the recovery of a client's files to occur, the index files must be stored within some storage, which is accessible for copying when needed (such as within the server or client). Furthermore, it is well known in the art that operating systems have the capability to verify files and the file system).

37. With regards to claim 66, Chang teaches a method wherein said index file is compressed (File compression requires a file to be rewritten in a different format. Chang's design discloses that an ID (a file) can be encoded (column 2, line 55, Chang). Encoding like compression requires a file to be rewritten in a different format. It would therefore have been inherent to use file compression).

38. With regards to claim 67, Chang teaches a method wherein files missing from said file system are retrieved from said server computer (Chang's design allows for workstation crash recovery (column 3, line 31, Chang). Workstation crash recovery is a process by which missing files are restored to the client machine by the server management application within the server. Hence missing files are retrieved from the server to the client as claimed).

39. With regards to claim 68, Chang teaches a method wherein said index file corresponds to said attributes of said client computer (Index files are used as a reference to maintain the correctness of files. In this case the index file serves as a

reference to files in client machines. If files serve as index files to a client, it is inherent that the files would possess not only information pertaining to the files within the client but also possess information pertaining to the client machine itself. The claim is thus rejected).

40. With regards to claim 69, Chang teaches a method wherein said files are retrieved using the PXE TFTP protocol (Chang's design teaches that the server management application can perform pre-boot tasks on the client machine (column 2, lines 65-66, Chang). To perform any task, a type of environment must exist (even for pre-boot tasks). Since the PXE TFTP protocol is a pre-boot execution environment, it is viewed as being equivalent to Chang's design which is capable of performing (executing) pre-boot tasks).

41. With regards to claim 75, Chang teaches a method comprising the step of mounting a remote volume of said server computer on said client computer (Chang's design calls for both server and client machines. The design enables the client and server to communicate between each other (column 4, lines 23-24, Chang). It is inherent that both the server and client machines have drives such as hard drives. Since a client can communicate with a server (which is remote), it is possible for a client to access a server's hard drive. This accessing of a server's hard drive requires mounting and hence the claimed ability to mount a remote drive from the server to the client is viewed as being inherent).

42. With regards to claim 76, Chang teaches a method wherein the step of executing said management instructions comprises accessing files stored on said remote volume (As stated before, Chang's design allows for a server management application within a server. It is inherent that a server has hard drives (which is a type of drive) by which to store data files. This setup is equivalent to the claimed data files located on a remote drive. It was also stated before that Chang's design allows for clients and servers to communicate with one another (column 4, lines 23-24, Chang). This arrangement permits the data files within the server (remote drive) to be accessed as claimed).

43. With regards to claim 77, Chang teaches a method wherein said client computer comprises a file system and wherein said step of managing said client computer comprises verifying said file system of said client computer (Operating systems contain file systems and Chang's design calls for client machines to have operating systems (column 2, line 20, Chang). In addition, Chang's design allows for workstation crash recovery (column 3, line 31, Chang). For workstation crash recovery to function properly, files within the client must be checked against some form of index of files. Hence the claimed verification process occurs).

44. With regards to claim 78, Chang teaches a method wherein files missing from said file system are retrieved from said remote volume (Chang's design allows for workstation crash recovery (column 3, line 31, Chang). Workstation crash recovery is a

process by which missing files are restored to the client machine by the server management application within the server. Hence missing files are retrieved from the file system to the client as claimed).

45. With regards to claim 79, Chang teaches a method wherein said step of verifying said file system comprises checking the files in said file system against an index file (Operating systems contain file systems and Chang's design calls for client machines to have operating systems (column 2, line 20, Chang). In addition, Chang's design allows for workstation crash recovery (column 3, line 31, Chang). For workstation crash recovery to function properly, files within the client must be checked against some form of index of files. Hence the claimed index file must exist).

46. With regards to claims 80-82, 86, and 89, Chang teaches a computer readable medium having instructions stored thereon for executing methods of claims 42, 44, 49, 68 and 76 (Chang's design features a hardware component with memory, such as ROM (a computer-readable medium) which is attached to the client machines to permit server management applications within servers to perform management tasks even during the pre-boot stage of the computer's runtime (column 2, lines 38-54, Chang)).

47. With regards to claim 90, Chang teaches a system comprising:

- A database configured to store a plurality of records of information, each of said plurality of records corresponding to one of said plurality of client

computers (Chang discloses a design which features a database within the server for holding the access control list (ACL) (column 4, line 3, Chang). If the database can hold an access control list, the access control list can contain information concerning the client as claimed. In addition, ACLs typically do contain multiple records);

- A server application configured to receive a request from one of said plurality of client computers via said network and to associate said request with said record corresponding to said one of said client computers based upon attributes of the one of said plurality of client computers (As stated above, Chang's design provides for communication between the server and the client machine (column 4, lines 23-24, Chang). This communication allows for the server to receive requests from the clients with information from the client (such as records) as claimed); and
- A plurality of configuration scripts each comprising instructions to be executed by at least one of said plurality of client computers, wherein one of said plurality of configuration scripts identified by the record corresponding to said one of said plurality of client computers is provided to said one of said plurality of client computers via said network by said server application in response to said request (Chang's design allows for the use of scripts to create utilities to perform administrative tasks (column 4, lines 33-35, Chang). The claimed tasks of configurations are administrative tasks. In addition, the utilities made from scripts are run on client machines and are sent through the

communication (network) established between the client and server (column 4, lines 23-24, Chang)).

48. With regards to claim 98, Chang teaches a system wherein said records of information comprise attributes of said client computers (As stated above, Chang's design consists of a server management application. For such an application to function properly, it is inherent that it not only contains the claimed management instructions, but it also should select the appropriate management instructions for each client machine. If this step did not exist, a server management application could never properly perform its task).

49. With regards to claim 99, Chang teaches a system wherein said attributes comprise DMI attributes (Chang's design has client machines with NICs to allow for interfacing with desktop server management applications (column 4, lines 13-15, Chang). Furthermore, Chang's design consists of a hardware device attached to each client machine to permit access to the client from the desktop management software at any given time (column 2, lines 38-54, Chang). These can be accessed by the server management application of Chang's design hence, the attributes can comprise of DMI attributes as claimed).

50. With regards to claim 100, Chang teaches a system wherein said attributes comprise PCI attributes (PCI is a type of hardware and Chang's design has client

machines with hardware (column 4, lines 10-12, Chang). These can be accessed by the server management application of Chang's design hence, the attributes can comprise of PCI attributes as claimed).

51. With regards to claim 101, Chang teaches a system wherein said attributes comprise SMBIOS attributes (SMBIOS is a type of BIOS. Chang's design allows for the server management application within a server to access a client's BIOS (column 2, line 48, Chang). For the server management application to access the BIOS properly, it must have some means by which to obtain the attributes of the BIOS. Since SMBIOS and BIOS are viewed as being equivalent, the server management application must be able to obtain the SMBIOS attributes).

52. With regards to claim 105, Chang teaches a system for administering a plurality of client computers over a network, the system comprising;

- Means for receiving a boot message from one of said plurality of client computers (Chang discloses a design that allows for server and client machines to communicate with each other (column 4, lines 19-24, Chang). Chang further discloses in his design that the server management application within the server can offer pre-boot updates and allow the client to access the server while the client is booting (column 2, lines 38-67, Chang). With such abilities available in the design, it is inherent that the server management

application within the server is able to detect when the client is booting and thus is able to receive boot messages);

- Means for recognizing said one of said plurality of client computers from said boot message (Chang discloses a design that allows for server and client machines to communicate with each other (column 4, lines 19-24, Chang). Chang further discloses in his design that the server management application within the server can offer pre-boot updates and allow the client to access the server while the client is booting (column 2, lines 38-67, Chang). With such abilities available in the design, it is inherent that the server management application within the server is able to detect when the client is booting and thus is able to receive boot messages. It is well known in the art that messages are simply data. The claimed received boot messages are able to contain data about the client booting. If boot messages are received then it is also inherent that the server management application should be able to detect which client machine the boot message is from (since such information will be contained within the message).
- Means for associating said boot message with an entry in a database to determine administration steps to be performed on said one of said plurality of client computers (Chang discloses a design that allows for server and client machines to communicate with each other (column 4, lines 19-24, Chang). Chang further discloses in his design that the server management application within the server can offer pre-boot updates and allow the client to access the

server while the client is booting (column 2, lines 38-67, Chang). With such abilities available in the design, it is inherent that the server management application within the server is able to detect when the client is booting and thus is able to receive boot messages. In addition, Chang discloses that the server with the server management application has a database. A database stores data in some order and hence has entries. It is thus inherent that the boot messages can be associated with entries within the database); and

- Means for providing said administrative steps to said one of said plurality of client computers in response to said boot messages (As stated before, Chang's design allows for boot messages and client computers. The server in Chang's design contains a server management application that is used for performing administrative tasks on client machines. Since boot messages can be received, it is inherent that the administrative steps can be performed in response to boot messages).

53. With regards to claim 106, Chang teaches a system further comprising means for determining attributes of one of said plurality of client computers (Chang's design permits the server with the server management application to meter and diagnose client machines (column 3, lines 31-33, Chang). Having the ability to diagnose and meter clients is equivalent to having the ability to determine a client's attributes).

54. With regards to claim 107, Chang teaches a system wherein said determining means comprises means for querying hardware and software attributes of one of said plurality of client computers (Chang's design permits the server with the server management application to meter and diagnose client machines (column 3, lines 31-33, Chang). Having the ability to diagnose and meter clients is equivalent to having the ability to determine a client's attributes).

55. With regards to claim 108, Chang teaches a system wherein said querying means comprises means for querying DMI parameters of one of said plurality of client computers (Chang's design has the means by which to monitor nodes, meter and diagnose software, and manage a client machine's configuration (column 3, lines 12-34, Chang). To perform such tasks, the means by which to query the client about this information must exist. Since Chang's design has the means by which to query a client, it is inherent that Chang's design also has the means by which to query a client's DMI parameters).

56. With regards to claim 109, Chang teaches a system wherein said querying means comprises means for querying PCI parameters of one of said plurality of client computers (Chang's design allows for a client machine's hard drive to be accessed remotely from the server management application (column 3, lines 21-22, Chang). To properly access the hard drive, the server management application must have means by which to query the hard drive. The hard drive like the PCI is a form of hardware.

Since Chang's design allows for the querying of a hard drive, it is inherent that is must also enable the means by which to query PCI parameters).

57. With regards to claim 110, Chang teaches a system wherein said querying means comprises means for querying SMBIOS parameters of said one of said plurality of client computers (Chang's design allows the server management application within a server to provide pre-boot services (column 4, lines 47-67). Chang's design also allows the server management application within a server to access the client's hard drive, meter and diagnose software and manage configurations (column 3, lines 12-34, Chang). To properly perform the tasks described, it is inherent that the means by which to query a client's BIOS. A SMBIOS is a form of BIOS and thus, Chang's design is considered to allow for the querying of a client's SMBIOS attributes).

58. With regards to claim 111, Chang teaches a method of maintaining files on a client computer comprising the steps of:

- Receiving a boot request at a server computer from said client computer (Chang discloses a design that allows for server and client machines to communicate with each other (column 4, lines 19-24, Chang). Chang further discloses in his design that the server management application within the server can offer pre-boot updates and allow the client to access the server while the client is booting (column 2, lines 38-67, Chang). With such abilities available in the design, it is inherent that the server management application

within the server is able to detect when the client is booting and thus is able to receive boot messages);

- Providing a response to said boot request from said server to said client, wherein said response comprises a file checking program configured to be executed on said client computer (As stated before, Chang's design allows for boot messages and client computers. The server in Chang's design contains a server management application that is used for performing administrative tasks on client machines. The server management application within the server is capable of receiving boot messages. Since boot messages can be received, it is inherent that the administrative steps can be performed in response to boot messages. In addition, Chang discloses that his design is capable of providing crash recovery (column 3, line 30, Chang). To properly perform the task of crash recovery, it is inherent that a file-checking program be used to first detect if anything is missing. The claim is therefore rejected);
- Receiving an index of files on said client computer from said file checking program (As stated above, Chang's design allows for workstation crash recovery. To perform the task of crash recovery properly, it is inherent that some index of files be available to properly detect if files are missing. It would further be inherent that such an index of files be received on the client machine);
- Providing updated files from said server to said client computer based upon said index (As stated above, Chang's design allows for workstation crash

recovery. To perform the task of crash recovery properly, it is inherent that updated files be sent from the server to the client based upon an index of files).

59. With regards to claim 112, Chang teaches a method comprising the step of mounting a volume of said server to said client computer (Chang's design calls for both server and client machines. The design enables the client and server to communicate between each other (column 4, lines 23-24, Chang). It is inherent that both the server and client machines have drives such as hard drives. Since a client can communicate with a server (which is remote), it is possible for a client to access a server's hard drive. This accessing of a server's hard drive requires mounting and hence the claimed ability to mount a remote drive from the server to the client is viewed as being inherent).

60. With regards to claim 113, Chang teaches a method wherein said volume is mounted via a network stack located in a ROM on said client computer (Chang's design allows for a server and client to communicate amongst each other (this is a network). As stated before, Chang's design allows the client to mount a volume on the server. The claimed stack in ROM is inherent since various tasks apply stacks in the computing world and a ROM is simply a means for storage of the stack. The task of mounting a volume as claimed would qualify as one of the tasks that would inherently use a stack and ROM, hence the claim is rejected)

recovery. To perform the task of crash recovery properly, it is inherent that updated files be sent from the server to the client based upon an index of files).

59. With regards to claim 112, Chang teaches a method comprising the step of mounting a volume of said server to said client computer (Chang's design calls for both server and client machines. The design enables the client and server to communicate between each other (column 4, lines 23-24, Chang). It is inherent that both the server and client machines have drives such as hard drives. Since a client can communicate with a server (which is remote), it is possible for a client to access a server's hard drive. This accessing of a server's hard drive requires mounting and hence the claimed ability to mount a remote drive from the server to the client is viewed as being inherent).

60. With regards to claim 113, Chang teaches a method wherein said volume is mounted via a network stack located in a ROM on said client computer (Chang's design allows for a server and client to communicate amongst each other (this is a network). As stated before, Chang's design allows the client to mount a volume on the server. The claimed stack in ROM is inherent since various tasks apply stacks in the computing world and a ROM is simply a means for storage of the stack. The task of mounting a volume as claimed would qualify as one of the tasks that would inherently use a stack and ROM, hence the claim is rejected)

61. With regards to claim 114, Chang teaches a method wherein said ROM is a ROM on a network interface card of said client computer (Chang's design features a hardware component that is installed on the LAN card. A LAN card is a network interface card (NIC). The hardware component can hold programs and perform permits and helps execute some of the administrative tasks from the server management application within the server. Chang further discloses that the hardware component has ROM as claimed).

62. With regards to claim 115, Chang teaches a method wherein said ROM is a PXE-enabled ROM (PXE-enabled ROM is beneficial for its pre-boot functionality. Chang's design features ROM but also describes how the design can perform administrative tasks prior to loading of the operating system (column 2, lines 46-47, Chang). This functionality is equivalent to that of the PXE-enabled ROM. The claim is therefore rejected).

Claim Rejections - 35 USC § 103

63. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the

invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

64. Claims 21-40, 50, 52-61, 70-74, 92-94 and 116 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chang in view of Havekost et al (Pat No. US005768119A), referred to hereafter as Havekost. Chang discloses a design with server and client machines that can communicate between one another (column 4, lines 23-24, Chang). In addition, Chang teaches how the client files can be checked by the server management application within the server. (In column 7, line 47-53, Chang discloses that the client can be checked to ensure the software is up to date. Software is made of files or could consist of only one file; hence software and files are viewed as being equivalent. Additionally, Chang states in column 3, lines 30-32 how his design allows for workstation recovery, metering and diagnostics; hence the design allows for checks, recovery, metering and diagnostics of files.) Such capabilities make it inherent that an index of some form must exist to enable client files to be checked and recovered. Chang however fails to specify the role of registries within his design.

65. In the same field of endeavor, Havekost discloses a design that uses registries (column 31, line 46-48, Havekost), event objects (object that handle events are viewed as being equivalent to event objects, column 31, lines 53-55, Havekost), template objects (a template that serves as a template for an object is viewed as being equivalent to a template object, column 9, lines 21-22, Havekost) and workstation objects (an attribute object for a client is viewed as being equivalent to a workstation object, column 25, lines 4-10, Havekost). Such objects and registries disclosed can be applied as the objects and registries claimed.

66. Accordingly it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang).

67. With regards to claim 21, Chang teaches through Havekost a method where said client computer comprises a registry file and wherein the method further comprises the step of verifying said registry file of said client computer (As stated above, Chang teaches a design where server and client machines can communicate with one another. Also previously stated, Chang's design teaches how the client files can be checked by the server management application within the server. Chang's design however fails to describe the checking of the registry file.

Havekost's design discloses the use of a registry database (column 31, line 47, Havekost). A database stores files and hence the existence of registry files is taught. Registry files exist within computers, and both clients and servers are computers.

Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to permit a server to verify the registry files of a client, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation

available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

68. With regards to claim 22, Chang teaches through Havekost a method wherein said step of verifying said registry file comprises checking entries in said registry file against a registry index file (As previously stated, operating systems contain file systems and Chang's design calls for client machines to have operating systems (column 2, line 20, Chang). In addition, Chang's design allows for workstation recovery (column 3, line 31, Chang). For workstation crash recovery to function properly, files within the client must be checked against some form of index of files. Hence the claimed index file must exist. Chang's design however fails to describe the checking of the registry file.

Havekost's design discloses the use of a registry database (column 31, line 47, Havekost). A database stores files and hence the existence of registry files is taught. Registry files exist within computers, and both clients and servers are computers.

Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to permit a server to verify a registry file against a registry index file, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

69. With regards to claim 23, Chang teaches through Havekost a method wherein said registry index file is retained on said server computer and wherein said step of verifying said registry file is executed on said server computer (As stated previously, operating systems contain file systems and Chang's design calls for client machines to have operating systems (column 2, line 20, Chang). In addition, Chang's design allows for workstation crash recovery (column 3, line 31, Chang). For workstation crash recovery to function properly, it is inherent that files within the client must be checked against some form of index of files. For the recovery of a client's files to occur, the index files must be stored within some storage, which is accessible for copying when needed (such as within the server or client). Furthermore the server management application within the server handles the process of installation, distribution, metering and diagnostics (column 3, lines 31-32, Chang) (as stated above, software is comprised of files and hence processes performed on software can be performed on files). Metering and diagnostics are viewed as being equivalent to verification and hence, the verification of the file system are performed on the server of Chang's design as claimed. Chang's design however fails to describe the checking of the registry file.

Havekost's design discloses the use of a registry database (column 31, line 47, Havekost). A database stores files and hence the existence of registry files is taught. Registry files exist within computers, and both clients and servers are computers.

Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to have both

the registry index located at and the verification executed at the server, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

70. With regards to claim 24, Chang teaches through Havekost a method wherein said registry index file is retained on said client computer and wherein said step of verifying said registry file is executed on said client computer (As previously stated, operating systems contain file systems and Chang's design calls for client machines to have operating systems (column 2, line 20, Chang). In addition, Chang's design allows for workstation crash recovery (column 3, line 31, Chang). For workstation crash recovery to function properly, it is inherent that files within the client must be checked against some form of index of files. For the recovery of a client's files to occur, the index files must be stored within some storage, which is accessible for copying when needed (such as within the server or client). Furthermore, it is well known in the art that operating systems have the capability to verify files and the file system. Chang's design however fails to describe the checking of the registry file.

Havekost's design discloses the use of a registry database (column 31, line 47, Havekost). A database stores files and hence the existence of registry files is taught. Registry files exist within computers, and both clients and servers are computers.

Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to have both the registry index located at and the verification executed at the client, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

71. With regards to claim 25, Chang teaches through Havekost a method wherein said registry index file corresponds to said attributes of said client computer (Index files are used as a reference to maintain the correctness of files. In this case the index file serves as a reference to files in client machines. If files serve as index files to a client, it is inherent that the files would possess not only information pertaining to the files within the client but also possess information pertaining to the client machine itself. Chang's design however fails to describe the checking of the registry file.

Havekost's design discloses the use of a registry database (column 31, line 47, Havekost). A database stores files and hence the existence of registry files is taught. Registry files exist within computers, and both clients and servers are computers.

Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to have the registry index file correspond to the said attributes of the said client computer, for the purpose of creating a method and apparatus for maintaining a computer system to

enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

72. With regards to claim 26, Chang teaches through Havekost a plurality of said data structures comprising a plurality of a workstation objects (Havekost discloses a design featuring attribute objects created by clients (column 25, lines 7-8, Havekost). These objects are viewed as being equivalent to workstation objects), each of said workstation objects corresponding to one of a plurality of workstations,

- Wherein each of said plurality of workstation objects comprises a plurality of data structures representing attributes of said one of said plurality of workstations, and wherein each of said workstation objects is associated with at least one of a plurality of template objects (As stated before, Chang's design has server management applications which permit administrators to meter, diagnose and recover files (also stated earlier, software is made of files and the two are considered equivalent) within client machines (column 3, lines 30-32, Chang). To properly perform these tasks, it is inherent that the attributes of the clients being worked on should be accessible to the server management application and that the attributes must exist in the form of files. In addition, files can be data structures and hence the two are viewed as being the same. Since attributes (in the form of files) are accessible to the server management application, Chang's design contains attribute files (data

structures). Chang's design however fails to disclose the use of workstation and template objects.

Havekost discloses a design which uses workstation objects (column 25, lines 7-8, Havekost) and template objects (a template for an object can be a template object) (column 9, lines 21-22, Havekost).

It would have been obvious to one skilled in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a design using data structures comprising a plurality of workstation objects comprising a plurality of data structures representing attributes of a plurality of workstations and each workstation is associated with a plurality of template objects, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)),

- Wherein each of said plurality of template objects is associated with one of a plurality of event objects (As stated above, Chang's design discloses how attributes of workstations can be accessed by the server management application within the server. Chang's design however fails to disclose the relationship between template objects with event objects.

Havekost discloses a design which uses event objects (an object that deals with events is interpreted as being an event object, column 31, lines 53-

55, Havekost) and template objects (a template for an object is interpreted as being a template object, column 9, lines 21-22, Havekost).

It would have been obvious to one skilled in the art at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a design associating template objects with event objects, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)),

- Wherein each of said plurality of event objects is associated with one of a plurality of script objects (As stated above, Chang's design discloses how attributes of workstations can be accessed by the server management application within the server. Chang further discloses how scripts can be used to allow the administrator to perform desired operations (scripts can be objects hence, script objects are viewed as being equivalent to scripts). Chang's design however fails to disclose the relationship between script objects with event objects.

Havekost discloses a design which uses event objects (an object that deals with events is interpreted as being an event object, column 31, lines 53-55, Havekost).

It would have been obvious to one skilled in the art at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a design where event objects are associated with script objects, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)), and

- Wherein each of said script objects comprises instructions to be executed by one of said plurality of workstations as part of a pre-boot sequence (As stated above, Chang's design discloses how attributes of workstations can be accessed by the server management application within the server. Chang further discloses how scripts can be used to allow the administrator to perform desired operations (scripts can be objects hence, script objects are viewed as being equivalent to scripts). To perform these operations, it is inherent that the scripts (script objects) execute instructions within a computer, such as a workstation. In addition, Chang discloses in his design that administrative tasks (such as file updates, scripts can be objects contained in files) can be performed prior to the booting of the operating system (column 4, lines 60-63, Chang). Chang's design however fails to disclose the relationship between script objects with event objects.

Havekost discloses a design which uses event objects (an object that deals with events is interpreted as being an event object, column 31, lines 53-55, Havekost).

It would have been obvious to one skilled in the art at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a design where script objects comprises instructions to be executed by workstation as part of a pre-boot sequence, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

73. With regards to claim 27, Chang teaches through Havekost that at least one workstation group object representing a workstation group corresponding to said plurality of workstations (Chang discloses a design which allows for scripting languages to be used to program client-server operations (column 4, lines 33-35, Chang). Hence, Chang teaches the use of scripts. Chang's design however fails to describe the use of templates.

Havekost's design discloses the use of a workstation objects (In column 25, lines 7-8, Havekost describes an attribute object created by a client, this is viewed as being equivalent to a workstation object). If an object can represent one workstation, it is

inherent that an object can represent multiple workstations. A workstation object is thus viewed as being equal to a workstation group object. It is also inherent that both servers and clients are computers and hence, these workstation objects can exist in either servers or clients.

Havekost discloses that workstations can have objects associated with them. Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to have a workstation group object represent a workstation group corresponding to a plurality of workstations, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

74. With regards to claim 28, Chang teaches through Havekost a medium wherein at least one of said template objects is associated with at least one workstation group object such that each of said plurality of workstations becomes associated with said at least one of said template objects via said at least one workstation group object objects (As stated before, Chang's design inherently allows for attributes. Chang's design however fails to disclose the use of workstation and template objects.

Havekost discloses a design which uses workstation objects (column 25, lines 7-8, Havekost) and template objects (a template for an object can be a template object) (column 9, lines 21-22, Havekost). An object is capable of representing a single or

multiple items hence; a workstation object can represent a single workstation or workstation groups. As stated before, a template object can be associated with a workstation group object. If such a relationship occurs, then the workstations become associated with the template objects. This association can occur through a workstation group object.

It would have been obvious to one skilled in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a design where template objects are associated with workstation group objects such that workstations become associated with template objects via a workstation group object, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

75. With regards to claim 29, Chang teaches through Havekost a medium wherein said at least one of said template objects is associated with at least one workstation object as a function of said attributes of said at lest one workstation object (As stated before, Chang's design has server management applications which permit administrators to meter, diagnose and recover files (also stated earlier, software is made of files and the two are considered equivalent) within client machines (column 3, lines 30-32, Chang). To properly perform these tasks, it is inherent that the attributes of the clients being worked on should be accessible to the server management application

and that the attributes must exist in the form of files. Chang's design however fails to disclose the use of workstation and template objects.

Havekost discloses a design which uses workstation objects (column 25, lines 7-8, Havekost) and template objects (a template for an object can be a template object) (column 9, lines 21-22, Havekost). As stated before, a relationship can exist between template objects and workstation objects. Objects hold data and attributes are a form of data hence; workstation objects can have attributes.

It would have been obvious to one skilled in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a design where template objects are associated with workstation objects as a function of attributes of workstation objects, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

76. With regards to claim 30, Chang teaches through Havekost a computer readable medium wherein said attributes comprise hardware attributes (Chang's discloses a design that features client machines with hardware (column 4, lines 10-12, Chang). These can be accessed by the server management application of Chang's design. Since the server management application can access the client machine, some attributes of the client must be known. These attributes can comprise of hardware

attributes as claimed. Chang's design however fails to disclose the use of workstation and template objects.

Havekost discloses a design which uses workstation objects (column 25, lines 7-8, Havekost) and template objects (a template for an object can be a template object) (column 9, lines 21-22, Havekost).

It would have been obvious to one skilled in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a design where attributes comprise of hardware attributes, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

77. With regards to claim 31, Chang teaches through Havekost a computer readable medium wherein said attributes comprise firmware attributes (Chang's design has client machines with firmware (column 4, lines 13-15, Chang). These can be accessed by the server management application of Chang's design hence, the attributes can comprise of firmware attributes as claimed. Chang's design however fails to disclose the use of workstation and template objects.

Havekost discloses a design which uses workstation objects (column 25, lines 7-8, Havekost) and template objects (a template for an object can be a template object) (column 9, lines 21-22, Havekost).

It would have been obvious to one skilled in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a design where attributes comprise of firmware attributes, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

78. With regards to claim 32, Chang teaches through Havekost a computer readable medium wherein said attributes comprise desktop management interface (DMI) attributes. (Chang's design has client machines with NICs to allow for interfacing with desktop server management applications (column 4, lines 13-15, Chang). Furthermore, Chang's design consists of a hardware device attached to each client machine to permit access to the client from the desktop management software at any given time (column 2, lines 38-54, Chang). These can be accessed by the server management application of Chang's design hence, the attributes can comprise of DMI attributes as claimed. Chang's design however fails to disclose the use of workstation and template objects.

Havekost discloses a design which uses workstation objects (column 25, lines 7-8, Havekost) and template objects (a template for an object can be a template object) (column 9, lines 21-22, Havekost).

It would have been obvious to one skilled in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a

design where attributes comprise of firmware attributes, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

79. With regards to claim 33, Chang teaches through Havekost a computer readable medium wherein said attributes are PCI attributes (PCI is a type of hardware and Chang's design has client machines with hardware (column 4, lines 10-12, Chang). These can be accessed by the server management application of Chang's design hence, the attributes can comprise of PCI attributes as claimed. Chang's design however fails to disclose the use of workstation and template objects.

Havekost discloses a design which uses workstation objects (column 25, lines 7-8, Havekost) and template objects (a template for an object can be a template object) (column 9, lines 21-22, Havekost).

It would have been obvious to one skilled in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a design where attributes comprise of firmware attributes, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

80. With regards to claim 34, Chang teaches through Havekost a computer readable medium wherein said attributes are SMBIOS attributes (SMBIOS is a type of BIOS. Chang's design allows for the server management application within a server to access a client's BIOS (column 2, line 48, Chang). For the server management application to access the BIOS properly, it must have some means by which to obtain the attributes of the BIOS. Since SMBIOS and BIOS are viewed as being equivalent, the server management application must be able to obtain the SMBIOS attributes. Chang's design however fails to disclose the use of workstation and template objects.

Havekost discloses a design which uses workstation objects (column 25, lines 7-8, Havekost) and template objects (a template for an object can be a template object) (column 9, lines 21-22, Havekost).

It would have been obvious to one skilled in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a design where attributes comprise of firmware attributes, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

81. With regards to claim 35, Chang teaches through Havekost a computer readable medium wherein said attributes comprise at least one of the group consisting of system

manufacturer, model, motherboard type, bus information, and adapter information (Chang's design allows for the server management application to access the motherboard (column 2, lines 38-44, Chang). Since the motherboard can be accessed, its attributes can be obtained hence, the motherboard type can be obtained. In addition for server management applications to function properly, attributes of the client must be obtained; any of the attributes claimed are reasonable attributes that could be claimed by many server management applications. Chang's design however fails to disclose the use of workstation and template objects.

Havekost discloses a design which uses workstation objects (column 25, lines 7-8, Havekost) and template objects (a template for an object can be a template object) (column 9, lines 21-22, Havekost).

It would have been obvious to one skilled in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a design where attributes comprise of firmware attributes, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

82. With regards to claim 36, Chang teaches through Havekost a computer readable medium wherein said adapter information comprises information about adapter orientation within a system bus of said client computer (For a server management

application within the server in Chang's design to function properly, it must know details concerning the client machines it is dealing with. Chang's disclosure states that an advantage of his design is the ability for "remote software installation, distribution, metering and diagnostics." In addition, it offers the ability for configuration management. Chang's disclosure further states the design offers "workstation recovery" and "critical node monitoring" (column 3, lines 25-33, Chang). With a design that allows for the server management application within the server to monitor such various aspects of the client, it is inherent that the client's adapter information (including that comprising information about the adapter's orientation within a system bus) would be available to the server management application as well. Chang's design however fails to disclose the use of workstation and template objects.

Havekost discloses a design which uses workstation objects (column 25, lines 7-8, Havekost) and template objects (a template for an object can be a template object) (column 9, lines 21-22, Havekost).

It would have been obvious to one skilled in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a design where attributes comprise of firmware attributes, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

83. With regards to claim 37, Chang teaches through Havekost a medium wherein said instructions comprise scripts executed by one of said plurality of workstations (Chang discloses a design which allows for scripting languages to be used to program client-server operations (column 4, lines 33-35, Chang). Client-server operations are viewed as being equivalent to management instructions. Chang's design however fails to disclose the use of workstation and template objects.

Havekost discloses a design which uses workstation objects (column 25, lines 7-8, Havekost) and template objects (a template for an object can be a template object) (column 9, lines 21-22, Havekost).

It would have been obvious to one skilled in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a design where attributes comprise of firmware attributes, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

84. With regard to claim 38, Chang teaches through Havekost a medium wherein said scripts are REXX scripts (Chang discloses a design which allows for scripting languages to be used to program client-server operations (column 4, lines 33-35, Chang). REXX is a scripting language. Chang's design however fails to disclose the use of workstation and template objects.

Havekost discloses a design which uses workstation objects (column 25, lines 7-8, Havekost) and template objects (a template for an object can be a template object) (column 9, lines 21-22, Havekost).

It would have been obvious to one skilled in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a design where attributes comprise of firmware attributes, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

85. With regards to claim 39, Chang teaches through Havekost a medium wherein said scripts are PERL scripts (Chang discloses a design which allows for scripting languages to be used to program client-server operations (column 4, lines 33-35, Chang). PERL is a scripting language. Chang's design however fails to disclose the use of workstation and template objects.

Havekost discloses a design which uses workstation objects (column 25, lines 7-8, Havekost) and template objects (a template for an object can be a template object) (column 9, lines 21-22, Havekost).

It would have been obvious to one skilled in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a design where attributes comprise of firmware attributes, for the purpose of creating a

method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

86. With regards to claim 40, Chang teaches through Havekost a medium wherein said scripts are batch scripts (Chang discloses a design which allows for scripting languages to be used to program client-server operations (column 4, lines 33-35, Chang). Chang stated no limitations and a batch script is still a form of a scripting language. Chang's design however fails to disclose the use of workstation and template objects.

Havekost discloses a design which uses workstation objects (column 25, lines 7-8, Havekost) and template objects (a template for an object can be a template object) (column 9, lines 21-22, Havekost).

It would have been obvious to one skilled in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a design where attributes comprise of firmware attributes, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

87. With regards to claim 50, Chang teaches through Havekost a method wherein each of said plurality of scripts is associated with a workstation object at said server, wherein said workstation object is associated with said client computer. (Chang discloses a design which allows for scripting languages to be used to program client-server operations (column 4, lines 33-35, Chang). Hence, Chang teaches the use of scripts. Chang's design however fails to describe the use of workstation objects.

Havekost's design discloses the use of a workstation objects (In column 25, lines 7-8, Havekost describes an attribute object created by a client, this is viewed as being equivalent to a workstation object). It is inherent that both servers and clients are computers and hence, these workstation objects can exist in either servers or clients.

The scripts are used to program client-server operations in Chang's design. Havekost discloses that workstations can have objects associated with them. It is therefore inherent that the scripts can be associated with the workstation objects. Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to have scripts associated with workstation objects, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

88. With regards to claim 52, Chang teaches through Havekost a method wherein at least one of said templates is associated with said script at said server through an event object. (Chang discloses a design which allows for scripting languages to be used to program client-server operations (column 4, lines 33-35, Chang). Hence, Chang teaches the use of scripts. Chang's design however fails to describe the use of templates and event objects.

Havekost's design discloses the use of templates (column 9, lines 21-22, Havekost) and event objects (column 31, line 54, Havekost).

The scripts are used to program client-server operations in Chang's design. The scripted operations can be created to perform in a desired way such as being associated with templates and event objects. Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to have scripts associated with templates through event objects (since the script could be configured that way), for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

89. With regards to claim 53, Chang teaches through Havekost a method wherein at least one of said templates is associated with said script at said server via a workstation group object. (Chang discloses a design which allows for scripting languages to be

used to program client-server operations (column 4, lines 33-35, Chang). Hence, Chang teaches the use of scripts. Chang's design however fails to describe the use of templates.

Havekost's design discloses the use of a workstation objects (In column 25, lines 7-8, Havekost describes an attribute object created by a client, this is viewed as being equivalent to a workstation object). If an object can represent one workstation, it is inherent that an object can represent multiple workstations. A workstation object is thus viewed as being equal to a workstation group object. It is also inherent that both servers and clients are computers and hence, these workstation objects can exist in either servers or clients. Havekost further discloses the use of templates in his design (column 9, lines 21-22, Havekost).

The scripts are used to program client-server operations in Chang's design. Havekost discloses that workstations can have objects associated with them. It is therefore inherent that the scripts can be associated with the templates via a workstation group object. Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to have scripts associated with workstation objects, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

90. With regards to claim 54, Chang teaches through Havekost a method wherein at least one of said templates is associated with said script at said server via said attributes of said client computer (Chang discloses a design which allows for scripting languages to be used to program client-server operations (column 4, lines 33-35, Chang). Hence, Chang teaches the use of scripts. In addition, it was stated earlier that Chang's design allows for the client machine's attributes to be accessed by the server management application within the server. Chang's design however fails to describe the use of templates.

Havekost's design discloses the use of templates (column 9, lines 21-22, Havekost).

The scripts are used to program client-server operations in Chang's design. In addition, Chang's design also teaches the use of client attributes. Havekost discloses the use of templates. It is therefore inherent that the scripts can be associated with the templates via the attributes of the client machine. Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to have scripts associated with workstation objects, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

91. With regards to claim 55, Chang teaches through Havekost a method wherein said attributes comprise hardware attributes (Chang's discloses a design that features client machines with hardware (column 4, lines 10-12, Chang). These can be accessed by the server management application of Chang's design. Since the server management application can access the client machine, some attributes of the client must be known. These attributes can comprise of hardware attributes as claimed. Chang's design however fails to describe the use of templates.

Havekost's design discloses the use of templates (column 9, lines 21-22, Havekost).

Chang's design teaches the use of client attributes. In addition, Havekost discloses the use of templates. It is therefore inherent that the hardware attributes can be associated with the templates via the attributes of the client machine. Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to have hardware attributes associated, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

92. With regards to claim 56, Chang teaches through Havekost a method wherein said attributes comprise at least one of the group consisting of manufacturer, model, motherboard type, bus information and adapter information (Chang's design allows for

the server management application to access the motherboard (column 2, lines 38-44, Chang). Since the motherboard can be accessed, its attributes can be obtained hence, the motherboard type can be obtained. In addition for server management applications to function properly, attributes of the client must be obtained; any of the attributes claimed are reasonable attributes that could be claimed by many server management applications. Chang's design however fails to describe the use of templates.

Havekost's design discloses the use of templates (column 9, lines 21-22, Havekost).

Chang teaches the use of attributes along with other computer information in his design. In addition, Havekost discloses the use of templates. It is therefore inherent that the information regarding manufacturer, model, motherboard type, bus information and adapter can be associated with the templates. Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to have hardware attributes associated, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

93. With regards to claim 57, Chang teaches through Havekost a method wherein said attributes comprise PCI attributes (PCI is a type of hardware and Chang's design has client machines with hardware (column 4, lines 10-12, Chang). These can be

accessed by the server management application of Chang's design hence, the attributes can comprise of PCI attributes as claimed. Chang's design however fails to describe the use of templates.

Havekost's design discloses the use of templates (column 9, lines 21-22, Havekost).

Chang teaches the use of PCI attributes along with other computer information in his design. In addition, Havekost discloses the use of templates. It is therefore inherent that PCI attributes can be associated with templates. Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to have hardware attributes associated, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

94. With regards to claim 58, Chang teaches through Havekost a method wherein said attributes are DMI attributes (Chang's design has client machines with NICs to allow for interfacing with desktop server management applications (column 4, lines 13-15, Chang). Furthermore, Chang's design consists of a hardware device attached to each client machine to permit access to the client from the desktop management software at any given time (column 2, lines 38-54, Chang). These can be accessed by the server management application of Chang's design hence, the attributes can

comprise of DMI attributes as claimed. Chang's design however fails to describe the use of templates.

Havekost's design discloses the use of templates (column 9, lines 21-22, Havekost).

Chang teaches the use of DMI attributes along with other computer information in his design. In addition, Havekost discloses the use of templates. It is therefore inherent that DMI attributes can be associated with templates. Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to have hardware attributes associated, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

95. With regards to claim 59, Chang teaches through Havekost a method wherein said attributes are SMBIOS attributes (SMBIOS is a type of BIOS. Chang's design allows for the server management application within a server to access a client's BIOS (column 2, line 48, Chang). For the server management application to access the BIOS properly, it must have some means by which to obtain the attributes of the BIOS. Since SMBIOS and BIOS are viewed as being equivalent, the server management application must be able to obtain the SMBIOS attributes. Chang's design however fails to describe the use of templates.

Havekost's design discloses the use of templates (column 9, lines 21-22, Havekost).

Chang teaches the use of SMBIOS attributes along with other computer information in his design. In addition, Havekost discloses the use of templates. It is therefore inherent that SMBIOS attributes can be associated with templates. Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to have hardware attributes associated, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

96. With regards to claim 60, Chang teaches through Havekost a method wherein said providing step and said receiving step are in accordance with the PXE protocol (Chang's design teaches that the server management application can perform pre-boot tasks on the client machine (column 2, lines 65-66, Chang). To perform any task, a type of environment must exist (even for pre-boot tasks). Since the PXE protocol is a pre-boot execution environment, it is viewed as being equivalent to Chang's design which is capable of performing (executing) pre-boot tasks. Chang's design however fails to describe the use of templates.

Havekost's design discloses the use of templates (column 9, lines 21-22, Havekost).

Chang teaches the use of a pre-boot execution environment in his design. In addition, Havekost discloses the use of templates. It is therefore inherent that pre-boot execution environments can be associated with templates. Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to have hardware attributes associated, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

97. With regards to claim 61, Chang teaches through Havekost a method wherein said providing step and said receiving step are substantially in accordance with the PXE protocol (Chang's design teaches that the server management application can perform pre-boot tasks on the client machine (column 2, lines 65-66, Chang). To perform any task, a type of environment must exist (even for pre-boot tasks). Since the PXE protocol is a pre-boot execution environment, it is viewed as being equivalent to Chang's design which is capable of performing (executing) pre-boot tasks. Chang's design however fails to describe the use of templates.

Havekost's design discloses the use of templates (column 9, lines 21-22, Havekost).

Chang teaches the use of a pre-boot execution environment in his design. In addition, Havekost discloses the use of templates. It is therefore inherent that pre-boot execution environments can be associated with templates. Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to have hardware attributes associated, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

98. With regards to claim 70, Chang teaches through Havekost a method wherein said client computer comprises a registry file and wherein said step of managing said client computer comprises verifying said registry file of said client computer (As stated above, Chang teaches a design where server and client machines can communicate with one another. Also previously stated, Chang's design teaches how the client files can be checked by the server management application within the server. Chang's design however fails to describe the checking of the registry file.

Havekost's design discloses the use of a registry database (column 31, line 47, Havekost). A database stores files and hence the existence of registry files is taught. Registry files exist within computers, and both clients and servers are computers.

Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to permit a

server to verify the registry files of a client, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

99. With regards to claim 71, Chang teaches through Havekost a method wherein said step of verifying said registry file comprises checking entries in said registry file against a registry index file (As previously stated, operating systems contain file systems and Chang's design calls for client machines to have operating systems (column 2, line 20, Chang). In addition, Chang's design allows for workstation recovery (column 3, line 31, Chang). For workstation crash recovery to function properly, files within the client must be checked against some form of index of files. Hence the claimed index file must exist. Chang's design however fails to describe the checking of the registry file.

Havekost's design discloses the use of a registry database (column 31, line 47, Havekost). A database stores files and hence the existence of registry files is taught. Registry files exist within computers, and both clients and servers are computers.

Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to permit a server to verify a registry file against a registry index file, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to

communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

100. With regards to claim 72, Chang teaches through Havekost a method wherein said registry index file is retained on said server computer and wherein said step of verifying said registry file is executed on said server computer (As stated previously, operating systems contain file systems and Chang's design calls for client machines to have operating systems (column 2, line 20, Chang). In addition, Chang's design allows for workstation crash recovery (column 3, line 31, Chang). For workstation crash recovery to function properly, it is inherent that files within the client must be checked against some form of index of files. For the recovery of a client's files to occur, the index files must be stored within some storage, which is accessible for copying when needed (such as within the server or client). Furthermore the server management application within the server handles the process of installation, distribution, metering and diagnostics (column 3, lines 31-32, Chang) (as stated above, software is comprised of files and hence processes performed on software can be performed on files). Metering and diagnostics are viewed as being equivalent to verification and hence, the verification of the file system are performed on the server of Chang's design as claimed. Chang's design however fails to describe the checking of the registry file.

Havekost's design discloses the use of a registry database (column 31, line 47, Havekost). A database stores files and hence the existence of registry files is taught. Registry files exist within computers, and both clients and servers are computers.

Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to have both the registry index located at and the verification executed at the server, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

101. With regards to claim 73, Chang teaches through Havekost a method wherein said registry index file is retained on said client computer and wherein said step of verifying said registry file is executed on said client computer (As previously stated, operating systems contain file systems and Chang's design calls for client machines to have operating systems (column 2, line 20, Chang). In addition, Chang's design allows for workstation crash recovery (column 3, line 31, Chang). For workstation crash recovery to function properly, it is inherent that files within the client must be checked against some form of index of files. For the recovery of a client's files to occur, the index files must be stored within some storage, which is accessible for copying when needed (such as within the server or client). Furthermore, it is well known in the art that

operating systems have the capability to verify files and the file system. Chang's design however fails to describe the checking of the registry file.

Havekost's design discloses the use of a registry database (column 31, line 47, Havekost). A database stores files and hence the existence of registry files is taught. Registry files exist within computers, and both clients and servers are computers.

Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to have both the registry index located at and the verification executed at the client, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

102. With regards to claim 74, Chang teaches through Havekost a method wherein said registry index file corresponds to said attributes of said client computer (Index files are used as a reference to maintain the correctness of files. In this case the index file serves as a reference to files in client machines. If files serve as index files to a client, it is inherent that the files would possess not only information pertaining to the files within the client but also possess information pertaining to the client machine itself. Chang's design however fails to describe the checking of the registry file.

Havekost's design discloses the use of a registry database (column 31, line 47, Havekost). A database stores files and hence the existence of registry files is taught. Registry files exist within computers, and both clients and servers are computers.

Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to have the registry index file correspond to the said attributes of the said client computer, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

103. With regards to claim 92, Chang teaches through Havekost a system wherein said records of information comprise template objects associated with each of said client computers (As stated before, Chang's design has server management applications which permit administrators to meter, diagnose and recover files (also stated earlier, software is made of files and the two are considered equivalent) within client machines (column 3, lines 30-32, Chang). To properly perform these tasks, it is inherent that the attributes of the clients being worked on should be accessible to the server management application and that the attributes must exist in the form of files. In addition, files can be data structures and hence the two are viewed as being equivalent. Since attributes (in the form of files) are accessible to the server management

application, Chang's design contains attribute files (data structures). Chang's design however fails to disclose the use of template objects.

Havekost discloses a design which uses template objects (a template for an object can be a template object) (column 9, lines 21-22, Havekost). It would have been inherent to associate such a template object with a workstation (which is a client machine).

It would have been obvious to one skilled in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a system that uses template objects associated with each of said client computers, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

104. With regards to claim 93, Chang teaches through Havekost a system wherein said records of information further comprise event objects associated with said template objects, wherein said event objects are associated with said configuration scripts such that said instructions are provided to said client computers upon the occurrence of an event (As stated before, Chang's design has server management applications which permit administrators to meter, diagnose and recover files (also stated earlier, software is made of files and the two are considered equivalent) within client machines (column 3, lines 30-32, Chang). To properly perform these tasks, it is inherent that the attributes

of the clients being worked on should be accessible to the server management application and that the attributes must exist in the form of files (these are equivalent to the claimed records of information). Since attributes (in the form of files) are accessible to the server management application, Chang's design contains attribute files. Furthermore, Chang discloses that his design enables the usage of scripts (column 4, lines 33-35, Chang). Chang's design however fails to disclose the use of event and template objects.

Havekost discloses a design which uses event objects (column 31, lines 53-55, Havekost) and template objects (a template for an object can be a template object) (column 9, lines 21-22, Havekost). Objects can hold information, such as the records of information claimed.

It would have been obvious to one skilled in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a design that associates event objects with scripts so that the scripts respond to an occurrence of an event, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

105. With regards to claim 94, Chang teaches through Havekost a system wherein said event comprises the booting of one of said client computers (Chang's design

discloses how updated files can be transferred to the client by the server management application within the server (column 4, lines 47-67, Chang). As stated before, the server management application can use scripts that run whatever tasks there are programmed to. Hence, the scripts can be created to respond to an event. The booting of a client machine is an event and Chang's design describes how clients can boot (column 4 lines 47-67, Chang). Hence a script can respond to a boot process. Chang's design however fails to disclose the use of event and template objects.

Havekost discloses a design which uses event objects (column 31, lines 53-55, Havekost) and template objects (a template for an object can be a template object) (column 9, lines 21-22, Havekost). Objects can hold information, such as the records of information claimed.

It would have been obvious to one skilled in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a design that associates event objects with scripts so that the scripts respond to an occurrence of an event, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

106. With regards to claims 83, 84, 85, 87 and 88, Chang teaches through Havekost a method of a computer readable medium having instructions stored thereon for

executing methods of claims 56, 57, 59, 70 and 74 (Chang's design features a hardware component with memory, such as ROM (a computer-readable medium) which is attached to the client machines to permit server management applications within servers to perform management tasks even during the pre-boot stage of the computer's runtime (column 2, lines 38-54, Chang). Chang's design however fails to describe the use of templates.

Havekost's design discloses the use of templates (column 9, lines 21-22, Havekost).

Chang teaches the use of a computer readable medium. In addition, Havekost discloses the use of templates. Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to have computer readable medium having instructions stored thereon for executing methods, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

107. With regards to claim 116, Chang teaches through Havekost a method of maintaining a registry on a client computer comprising the steps of:

- Receiving a boot request at a server computer from said client computer (Chang discloses a design that allows for server and client machines to

communicate with each other (column 4, lines 19-24, Chang). Chang further discloses in his design that the server management application within the server can offer pre-boot updates and allow the client to access the server while the client is booting (column 2, lines 38-67, Chang). With such abilities available in the design, it is inherent that the server management application within the server is able to detect when the client is booting and thus is able to receive boot messages. Chang's design however fails to describe the role of the registry file (registry).

Havekost's design discloses the use of a registry database (column 31, line 47, Havekost). A database stores files and hence the existence of registry files is taught. Registry files exist within computers, and both clients and servers are computers.

Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to permit a server to receive a boot request from a client maintaining a registry, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang));

- Providing a registry checking program to the client computer in response to said boot request, wherein said registry checking program is configured to

check said registry on said client computer and to provide a registry response to said server computer. Chang's design however fails to describe the role of the registry file (registry).

Havekost's design discloses the use of a registry database (column 31, line 47, Havekost). A database stores files and hence the existence of registry files is taught. Registry files exist within computers, and both clients and servers are computers.

Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to permit a server to respond to a boot request from a client maintaining a registry, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang));

- Receiving said registry response at said server from said registry checking program (As stated before, Chang's design allows for boot messages and client computers. The server in Chang's design contains a server management application that is used for performing administrative tasks on client machines. The server management application within the server is capable of receiving boot messages. Since boot messages can be received, it is inherent that the administrative steps can be performed in response to

boot messages. In addition, Chang discloses that his design is capable of providing crash recovery (column 3, line 30, Chang). To properly perform the task of crash recovery, it is inherent that a registry-checking program be used to first detect if anything is missing. Chang's design however fails to describe the role of the registry file (registry).

Havekost's design discloses the use of a registry database (column 31, line 47, Havekost). A database stores files and hence the existence of registry files is taught. Registry files exist within computers, and both clients and servers are computers.

Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to permit a server to receive a registry program from a registry checking program, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang));

- Processing said registry response at said server to verify said registry on said client computer (As stated before, Chang's design allows for boot messages and client computers. The server in Chang's design contains a server management application that is used for performing administrative tasks on client machines. The server management application within the server is

capable of receiving boot messages. Since boot messages can be received, it is inherent that the administrative steps can be performed in response to boot messages. In addition, Chang discloses that his design is capable of providing crash recovery (column 3, line 30, Chang). To properly perform the task of crash recovery, it is inherent that the server be able to verify the registry. Chang's design however fails to describe the role of the registry file (registry).

Havekost's design discloses the use of a registry database (column 31, line 47, Havekost). A database stores files and hence the existence of registry files is taught. Registry files exist within computers, and both clients and servers are computers.

Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to permit a server verify the registry, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)); and

- Providing an updated registry from said server to said client computer in response to said processing step (As stated before, Chang's design allows for boot messages and client computers. The server in Chang's design contains

a server management application that is used for performing administrative tasks on client machines. The server management application within the server is capable of receiving boot messages. Since boot messages can be received, it is inherent that the administrative steps can be performed in response to boot messages. In addition, Chang discloses that his design is capable of providing crash recovery (column 3, line 30, Chang). To properly perform the task of crash recovery, it is inherent that the server provides an updated registry to the client. Chang's design however fails to describe the role of the registry file (registry).

Havekost's design discloses the use of a registry database (column 31, line 47, Havekost). A database stores files and hence the existence of registry files is taught. Registry files exist within computers, and both clients and servers are computers.

Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to permit a server to provide a updated registry to the client, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

108. Claim 91 is rejected under 35 U.S.C. 103(a) as being unpatentable over Chang in view of Sonderegger et al (U.S. Pat No: 5692129), referred to hereafter as Sonderegger. With regards to claim 91, Chang teaches through Sonderegger a system wherein said database is a directory services application (Chang's design features a database (column 4, line 3, Chang). Chang however fails to specify the role of directory services applications.

109. Sonderegger discloses a design featuring a directory services application (NDS is a directory services application, column 5, lines 14-15, Sonderegger). This directory services application can serve as a database.

110. Hence it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Chang with those of Sonderegger, to have a database that is a directory services application, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

111. Claims 41, 95-97 and 102-104 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chang in view of Havekost et al (Pat No. US005768119A), referred to hereafter as Havekost and in further view of Sonderegger et al (Pat No. US005692129A), referenced hereafter as Sonderegger. Chang discloses a design with server and client machines that can communicate between one another (column 4, lines

23-24, Chang). In addition, Chang teaches how the client files can be checked by the server management application within the server. (In column 7, line 47-53, Chang discloses that the client can be checked to ensure the software is up to date. Software is made of files or could consist of only one file; hence software and files are viewed as being equivalent. Additionally, Chang states in column 3, lines 30-32 how his design allows for workstation recovery, metering and diagnostics; hence the design allows for checks, recovery, metering and diagnostics of files.) Such capabilities make it inherent that an index of some form must exist to enable client files to be checked and recovered. Chang however fails to specify the role of objects and directory service applications within his design.

112. In the same field of endeavor, Havekost discloses a design that uses registries (column 31, line 46-48, Havekost), event objects (object that handle events are viewed as being equivalent to event objects, column 31, lines 53-55, Havekost), template objects (a template that serves as a template for an object is viewed as being equivalent to a template object, column 9, lines 21-22, Havekost) and workstation objects (an attribute object for a client is viewed as being equivalent to a workstation object, column 25, lines 4-10, Havekost). Havekost's design however fails to describe the role of directory services applications.

113. Sonderegger discloses a design which features a directory services application (NDS is a type of directory services application, column 5, lines 14-15, Sonderegger). Such a directory services application can serve as a database.

114. Accordingly it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost and Sonderegger, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang).

115. With regards to claim 41, Chang teaches through Havekost and Sonderegger a medium wherein each of said workstation objects are directory services objects (Chang design teaches that databases can be used (column 4, line 3, Chang). Chang however fails to describe the role of objects and directory services objects.

In the same field of endeavor, Havekost discloses a design that features event objects (column 31, lines 53-55, Havekost), template objects (column 9, lines 21-22, Havekost) and workstation objects (column 25, lines 4-10, Havekost). Havekost however does not disclose details about directory services objects.

Sonderegger discloses a design which features a directory services application (NDS is a type of directory services application, column 5, lines 14-15, Sonderegger). Objects can be used to represent items. A directory services object represents a directory services application. Such directory services applications can be located within workstations. Hence, workstation objects can represent directory services applications.

Accordingly it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost and Sonderegger, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang).

116. With regards to claim 95, Chang teaches through Havekost and Sonderegger a system wherein said database is a directory services application (Chang design teaches that databases can be used (column 4, line 3, Chang). Chang however fails to describe the role of objects and directory services applications.

In the same field of endeavor, Havekost discloses a design that features event objects (column 31, lines 53-55, Havekost), template objects (column 9, lines 21-22, Havekost) and workstation objects (column 25, lines 4-10, Havekost). Havekost however does not disclose details about a directory services application.

Sonderegger discloses a design which features a directory services application (NDS is a type of directory services application, column 5, lines 14-15, Sonderegger). Such a directory services application can serve as a database.

Accordingly it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost and Sonderegger, for the purpose of creating a method and apparatus for maintaining a

computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang).

117. With regards to claim 96, Chang teaches through Havekost and Sonderegger a system wherein said directory services application is a Netware Directory Services directory (Chang design teaches that databases can be used (column 4, line 3, Chang). Chang however fails to describe the role of objects and directory services applications.

In the same field of endeavor, Havekost discloses a design that features event objects (column 31, lines 53-55, Havekost), template objects (column 9, lines 21-22, Havekost) and workstation objects (column 25, lines 4-10, Havekost). Havekost however does not disclose details about a directory services application.

Sonderegger discloses a design which features Netware Directory Services (NDS is a type of directory services application, column 5, lines 14-15, Sonderegger). Such a directory services application can serve as a database.

Accordingly it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost and Sonderegger, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server

management application running on the server via the network (column 2, lines 49-52, Chang).

118. With regards to claim 97, Chang teaches through Havekost and Sonderegger a system wherein said directory services application is a Microsoft Active Directory directory (Chang design teaches that databases can be used (column 4, line 3, Chang). Chang however fails to describe the role of objects and directory services applications.

In the same field of endeavor, Havekost discloses a design that features event objects (column 31, lines 53-55, Havekost), template objects (column 9, lines 21-22, Havekost) and workstation objects (column 25, lines 4-10, Havekost). Havekost however does not disclose details about a directory services application.

Sonderegger discloses a design which features a directory services application (NDS is a type of directory services application, column 5, lines 14-15, Sonderegger). Such a directory services application can serve as a database. Microsoft Active Directory like NDS is a directory services application. Hence, the use of one can be substituted by the use of the other.

Accordingly it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost and Sonderegger, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server

management application running on the server via the network (column 2, lines 49-52, Chang).

119. With regards to claim 102, Chang teaches through Havekost and Sonderegger a system for managing a plurality of client computers over a network, the system comprising:

- A directory services application configured to store objects associated with each of said plurality of client computers; (Chang teaches a design that uses scripts and client computers as stated above. Chang's design however fails to describe the role of a directory services application and objects.

  Havekost discloses a design which teaches the use of event objects, template objects and client objects as stated above. Havekost however does not teach about a directory services application.

  Sonderegger discloses a design which features a directory services application. Such an application can serve as a database and save data such as objects as claimed.

  Havekost teaches how objects can be associated with client computers. Sonderegger discloses a directory services application. A directory services application can store data such as client objects. Hence it would have been obvious to one in the art at the time of the invention to have combined the teachings of Chang with those of Havekost in addition to those of Sonderegger, for the purpose of creating a method and apparatus for

maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang));

- A plurality of configuration scripts each comprising instructions to be executed by one of said client computers (Chang teaches a design that uses scripts and client computers as stated above. Chang's design however fails to describe the role of a directory services application and objects.

Havekost discloses a design which teaches the use of event objects, template objects and client objects as stated above. Havekost however does not teach about a directory services application.

Sonderegger discloses a design which features a directory services application. Such an application can serve as a database and save data such as objects.

Chang teaches a design that allows for the usage of scripts to perform desired tasks (column 4, lines 33-35, Chang). While these scripts are within the server management application in the server, the server and clients can communicate between one another as stated before and hence, the scripts can be transmitted to the client machines to be executed as claimed. Havekost teaches how objects can be associated with client computers. Sonderegger discloses a directory services application. A directory services application can store data such as client objects. Hence it would have been

obvious to one in the art at the time of the invention to have combined the teachings of Chang with those of Havekost in addition to those of Sonderegger, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang));

- An interface configured to allow an administrator to select one of the plurality of configuration scripts to be executed by one of said plurality of client computers upon the occurrence of an event (Chang teaches a design that uses scripts and client computers as stated above. Chang further discloses the role of a server management application within the server. The server management application allows an administrator to control various client machines hence; it is inherent that it must contain the claimed interface. It is also inherent that the administrator would be able to select a script through this interface since multiple scripts can exist. Finally, as stated before, it is inherent that scripts can respond to events as claimed. Chang's design however fails to describe the role of a directory services application and objects.

Havekost discloses a design which teaches the use of event objects, template objects and client objects as stated above. Havekost however does not teach about a directory services application.

Sonderegger discloses a design which features a directory services application. Such an application can serve as a database and save data such as objects.

Chang teaches a design allowing for an interface to select scripts which respond to events. Since, scripts can be transferred to client machines as stated above (the server and client machines can communicate between one another and thus the client can receive scripts from the server), it is inherent that the client machine can execute the scripts as claimed. Further, it would have been obvious to one in the art at the time of the invention to have combined the teachings of Chang, Havekost and Sonderegger for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)); and

- A server application configured to provide said selected script to said one of the plurality of client computers via said network in response to said occurrence of said event (Chang teaches a design that uses scripts and client computers as stated above. Chang further discloses the role of a server management application within the server. The server management application allows an administrator to control various client machines. As stated above, the client and server are able to communicate with one another,

teaching the use of a network. Since the server and client are able to communicate with one another and the server management application within the server can diagnose the clients (column 3, lines 31-32, Chang), it is inherent that the server can send a script to the client when an event is detected, as claimed. Chang's design however fails to describe the role of a directory services application and objects.

Havekost discloses a design which teaches the use of event objects, template objects and client objects as stated above. Havekost however does not teach about a directory services application.

Sonderegger discloses a design which features a directory services application. Such an application can serve as a database and save data such as objects.

Chang teaches the transmission of scripts from a server to a client in response to an event. In addition, Havekost and Sonderegger teach the role of objects and directory services applications. Therefore, it would have been obvious to one in the art at the time of the invention to have combined the teachings of Chang, Havekost and Sonderegger for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

120. With regards to claim 103, Chang teaches through Havekost and Sonderegger, a system wherein said event is associated to said one of said client computers by a template object in said directory services application (Chang teaches a design that uses scripts and client computers as stated above. Chang further discloses the role of a server management application within the server. The server management application allows an administrator to control various client machines. Also stated above, the client and server are able to communicate with one another. Chang's design however fails to describe the role of a directory services application and objects.

Havekost discloses a design which teaches the use of template objects, as stated above. It is well known in the art that an object can be made to be associated with any computer, hence the template object can be associate the client machine with an application. Havekost however does not teach about a directory services application.

Sonderegger discloses a design which features a directory services application. Such an application can serve as a database and be associated with data such as objects.

Chang teaches the transmission of scripts from a server to a client in response to an event. In addition, Havekost and Sonderegger teach the role of template objects and directory services applications. It would have been inherent to associate a client with a directory services application through a template object as claimed. Therefore, it would have been obvious to one in the art at the time of the invention to have combined the

teachings of Chang, Havekost and Sonderegger for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

121. With regards to claim 104, Chang teaches through Havekost and Sonderegger a system wherein at least two of said client computers are associated in said directory services application by a workstation group object (Chang teaches a design that uses scripts and client computers as stated above. Chang further discloses the role of a server management application within the server. The server management application allows an administrator to control various client machines. Also stated above, the client and server are able to communicate with one another. Chang's design however fails to describe the role of a directory services application and objects.

Havekost discloses a design which teaches the use of workstation objects, as stated above. It is well known in the art that an object can be made to be associated with any computer; hence the workstation object can be used to associate the client machine with other client machines. Havekost however does not teach about a directory services application.

Sonderegger discloses a design which features a directory services application. Such an application can serve as a database and be associated with data such as objects, this include workstation objects.

Chang teaches the transmission of scripts from a server to a client in response to an event. In addition, Havekost and Sonderegger teach the role of workstation objects and directory services applications. It would have been inherent to associate a client with other clients in a directory services application through workstation objects as claimed. Therefore, it would have been obvious to one in the art at the time of the invention to have combined the teachings of Chang, Havekost and Sonderegger for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

Response to Remarks

The arguments filed by the applicant on March 5, 2004 have been thoroughly considered but they are not deemed fully persuasive. The following are brief explanations in response to some of the arguments presented.

The applicant's representatives disputes that the Chang disclosure deals solely with the specialized hardware on the client-side. The examiner disagrees with this view for Chang's disclosure first deals with the hardware and firmware on the client-side. Firmware includes both hardware and software by definition. Second, Chang discloses administrative advantages provided by such a design (column 3, lines 34-50, Chang), such as file updates (column 4, lines 44-45, Chang). The purpose of the design along

with certain features is presented within Chang's disclosure, and they involve a method for providing network management. The fact that not all the particular features claimed are stated specifically is irrelevant due to the fact that as stated, those features are inherently present in a design such as this.

For instance, applicant's representatives state that Chang does not account for multiple non-identical workstations. Chang does account for them, it is simply not stated the same way. Chang's design allows for a LAN (Figure 1), this alone shows that multiple workstations are contemplated by Chang's design. In addition though, Chang goes on to recite "user workstations can be configured centrally," (column 3, line 40, Chang). Furthermore, Chang discloses how the design allows for authentication of client machines (column 2, lines 55-57, Chang). Plus, Chang states that the network management processes can be performed on the client machines despite their configurations and environments (column 3, lines 41-43, Chang). The client computers hence are not all identical and Chang's design does indeed contemplate multiple non-identical workstations.

With regards to the arguments that Chang's design does not contemplate the use of multiple sets of management instructions selected based upon attributes of the particular workstation, this too is disagreed upon by the examiner. As stated above, Chang's design allows for network managing on client computer despite the computers' configuration and environment. Those skilled in the art understand that the only way to perform such as task is by detecting the configuration of a client computer (equivalent to Chang's feature: metering (column 3, line 48, Chang)) and selecting from a multiple set

of instructions. First of all, if only a single instruction existed, there exists no means within the computing field by which to perform network-managing tasks. It is thus inherent that multiple sets of instructions must exist. Second, a selection of the appropriate set(s) of instructions needs to be made (since applying them all may indeed defeat the goal of managing and correcting errors in a network, and having simply one set of instructions does not provide a reasonable amount of solutions to the number of network problems); and the factors used to make such a decision always includes the attributes of the workstation to be dealt with. The fact that Chang states that the design accounts for remote software installation, distribution, metering and diagnostics makes it clearly inherent that means by which to determine workstation attributes are present. In addition, it is inherent that the server for the purpose of managing the network obtains such data. This means that the workstation attributes inherently are obtained for the purpose of selecting the appropriate management instructions.

Finally with regards to the arguments that Chang does not contemplate an application provided by the server, to the client, prior to the receiving step, this too is disagreed upon by the examiner. First, means by which to perform such a task are present in Chang's design. Remote software installation and distribution to client machines is present in Chang's design (column 3, lines 31-33, Chang). In addition, Chang further states that files can be transferred from the server to the workstation (column 2, lines 66-67, Chang). This just further shows that programs can be sent from the server to the workstation. It is inherent that an application must be provided by the server to the client prior to the receiving because there is no other way for a client to

receive data over a network. Anyone within the networking and computing field is informed of the fact that for data transfers to occur in a network (including the receipt of commands), both the server and the client, each must have an initial software and/or hardware to allow for such tasks. In Chang's design, this initial software/hardware (application) present within the client prior to the receiving step is the firmware.

Applicant's representatives argue that certain features are not expressly stated within the prior art disclosure. It is not possible for disclosures to detail every aspect of a design. It is assumed certain traits and features of a design will be inherent since operation of the design will not be successful without them.

The prior art disclosures should not be reviewed simply literally, the spirit of the design should be taken into account as well. It is strongly believed that the prior arts provided, especially the Chang disclosure demonstrates that the design claimed by the applicant is not unique.

It is appreciated that the applicant took the 112 rejections into consideration and that the appropriate changes were made. However, the trademark objections will continue to stand. When trademarked terms are used, it is setting an unclear limitation to the design. This become even more of a concern when trademarked terms are used within claim language. It creates a limitation based on another design which in turn makes the scope of the design ever more difficult to decipher. It is preferred that trademarked language not be used when it can be avoided to prevent such a lack of finiteness, but should it be necessary to use it, it is a strong belief within the office that

the TM emblem be applied to provide both clarification and to preserve the trademarked entity.

While not all the concerns have been addressed within this response, it is believed that the major ones have been. The examiner has devoted great time and thought towards this case and has consulted certain issues with others within the office. It is believed that the rejections made are just and that the explanations provided within this response will assist in further clarifying the reasons for the rejections.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Azizul Choudhury whose telephone number is 703-305-7209. The examiner can normally be reached on M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, David Wiley can be reached on 703-308-5221. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AC

DAVID WILEY
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100